

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

Área de formación: Disciplinaria

Unidad académica: Teoría Matemática de la Computación

Ubicación: Tercer Semestre

Clave: 1975

Horas semana-mes: 5

Horas teoría: 5

Horas práctica: 0

Unidades CONAIC: 80

Prerrequisitos: Ninguno

Horas de infraestructura: 0

Créditos: 10

PRESENTACIÓN

Las Matemáticas brindan una base de tipo formativo para el desarrollo de habilidades de abstracción, lógica y expresión de formalismos. Además proporcionan los conocimientos fundamentales para el enriquecimiento de la comprensión de la disciplina computacional.

La Teoría Matemática de la computación es considerada una asignatura básica para los estudiantes de informática. Dicha teoría modela matemáticamente máquinas simples con capacidades bien definidas y entrega las bases para determinar ciertos tipos de problemas que tienen solución, los que son bien caracterizados, y entrega elementos teóricos para determinar cuando un problema no tiene solución algorítmica.

OBJETIVO GENERAL

Al final del curso el alumno aplicará los conocimientos formales que sustentan al modelo teórico y conceptos de las computadoras en su sentido amplio.

UNIDAD I.- AUTÓMATAS

TIEMPO APROXIMADO: 25 Horas

OBJETIVO DE LA UNIDAD: Conocer la teoría de autómatas.

CONTENIDO

1.1 Máquinas finitas

1.1.1 Definiciones elementales: estados, símbolos, transiciones

1.1.2 Teoremas de equivalencia entre lenguajes producidos por gramáticas y lenguajes reconocidos por autómatas

1.1.3 Jerarquización de autómatas

1.1.3.1 Autómatas finitos

1.1.3.1.1 Autómata finito determinístico (AFD)

1.1.3.1.2 Autómata finito no determinístico (AFND)

1.1.3.1.3 Equivalencia de AFND y AFD

1.1.3.2 Autómatas de pila

UNIDAD II.- LENGUAJES FORMALES

TIEMPO APROXIMADO: 15 Horas

OBJETIVO DE LA UNIDAD: Conocer la teoría de lenguajes formales.

CONTENIDO

- 2.1 Reconocimiento de lenguajes
 - 2.1.1 Relaciones estructurales entre autómatas y gramáticas
 - 2.1.2 Generación de lenguajes: árboles de derivación
 - 2.1.3 El problema del reconocimiento
 - 2.1.4 Esquemas de análisis sintáctico: parsing ascendente y descendente
 - 2.1.5 Algoritmos de análisis sintáctico para lenguajes independientes del contexto.
- 2.2 Lenguajes Formales
 - 2.2.1 Alfabeto, cadenas, lenguajes y operaciones
 - 2.2.2 Gramáticas formales: definiciones, operaciones, tipos de lenguajes, ambigüedad, equivalencia, jerarquización de Chomsky.

UNIDAD III.- SISTEMAS FORMALES

TIEMPO APROXIMADO: 25 Horas

OBJETIVO DE LA UNIDAD: Aplicar los procedimientos de computabilidad y definir las funciones recursivas que permiten aprovechar el uso de las computadoras.

CONTENIDO

- 3.1 Máquinas de Turing
 - 3.1.1 Concepto de computabilidad
 - 3.1.2 Concepto de procedimientos, procedimiento efectivo y algoritmo
 - 3.1.3 Máquinas de Turing: modelos de computabilidad, problemas indecidibles (The Halting Problem)
 - 3.1.4 Límites de computabilidad
 - 3.1.5 Relaciones entre máquinas de Turing y teoría de funciones recursivas
 - 3.1.6 Equivalencias formales
- 3.2 Funciones recursivas
 - 3.2.1 Funciones computables y algoritmos
 - 3.2.2 Funciones recursivas

UNIDAD IV.- COMPUTABILIDAD

TIEMPO APROXIMADO: 15 Horas

OBJETIVO DE LA UNIDAD: Comprender la complejidad y acotamiento en teorías de primer orden para la computabilidad.

CONTENIDO

- 4.1 Complejidad
 - 4.1.1 Complejidad y computabilidad
 - 4.1.2 Complejidad de algoritmos
 - 4.1.3 Teorema del acotamiento
 - 4.1.4 Clases de complejidad
 - 4.1.5 Computabilidad polinomial

- 4.1.6 Clases de P y NP
- 4.1.7 Algoritmos NP
- 4.1.8 Problemas NP completos
- 4.1.9 Problema de la satisfabilidad
- 4.1.10 Problemas intratables
- 4.1.11 complejidad de teorías de primer orden

EXPERIENCIAS DE APRENDIZAJE

- Realización de problemas prácticos
- Demostración práctica de la aplicación de los conceptos de expresiones regulares en sistemas operativos

CRITERIOS DE EVALUACIÓN

Exámenes escritos	40%
Ejercicios en clase	10%
Investigación	20%
Proyecto Final	30%
TOTAL	<u>100%</u>

RECURSOS NECESARIOS

Recursos didácticos:
Aula, pizarrón, gis.

PRÁCTICAS SUGERIDAS

- Diseño de Autómatas Finitos a partir del planteamiento de problemas.
- Diseño de Gramáticas Regulares e Independientes del Contexto a partir del planteamiento de problemas.
- Usar un software para validar cadenas a partir de autómatas.
- Prácticas con herramientas computacionales adecuadas a los temas del curso.
- Desarrollo de programas para autómatas finitos no deterministas.
- Desarrollo de un analizador léxico básico.

BIBLIOGRAFÍA BÁSICA

GAUDIOSO, E. (2010). Autómatas, Gramáticas y Lenguajes Formales: Problemas Resueltos. Madrid: Sanz y Torres Editorial.
ALFONSECA, E., et al. (2007). Teoría de autómatas y lenguajes formales. Madrid: McGraw Hill.

BIBLIOGRAFÍA COMPLEMENTARIA

BROOKSHEAR, G. (1999). Teoría de la computación. México: Addison-Wesley.
HOPCROFT, J., et al. (1993). Introducción a la teoría de autómatas, lenguajes y computación. México: CECSA.
LINZ, P. (2001). An introduction to formal languages and automata. USA: Jones and Bartlett.